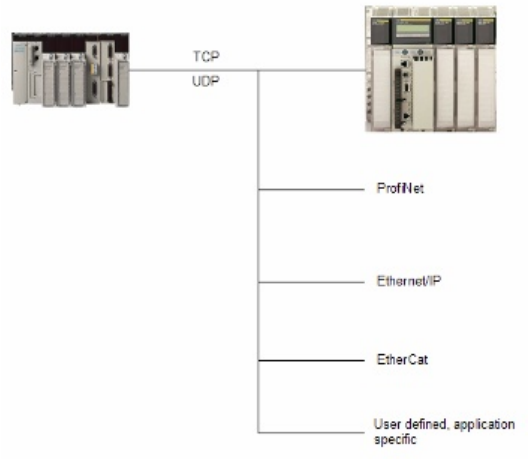
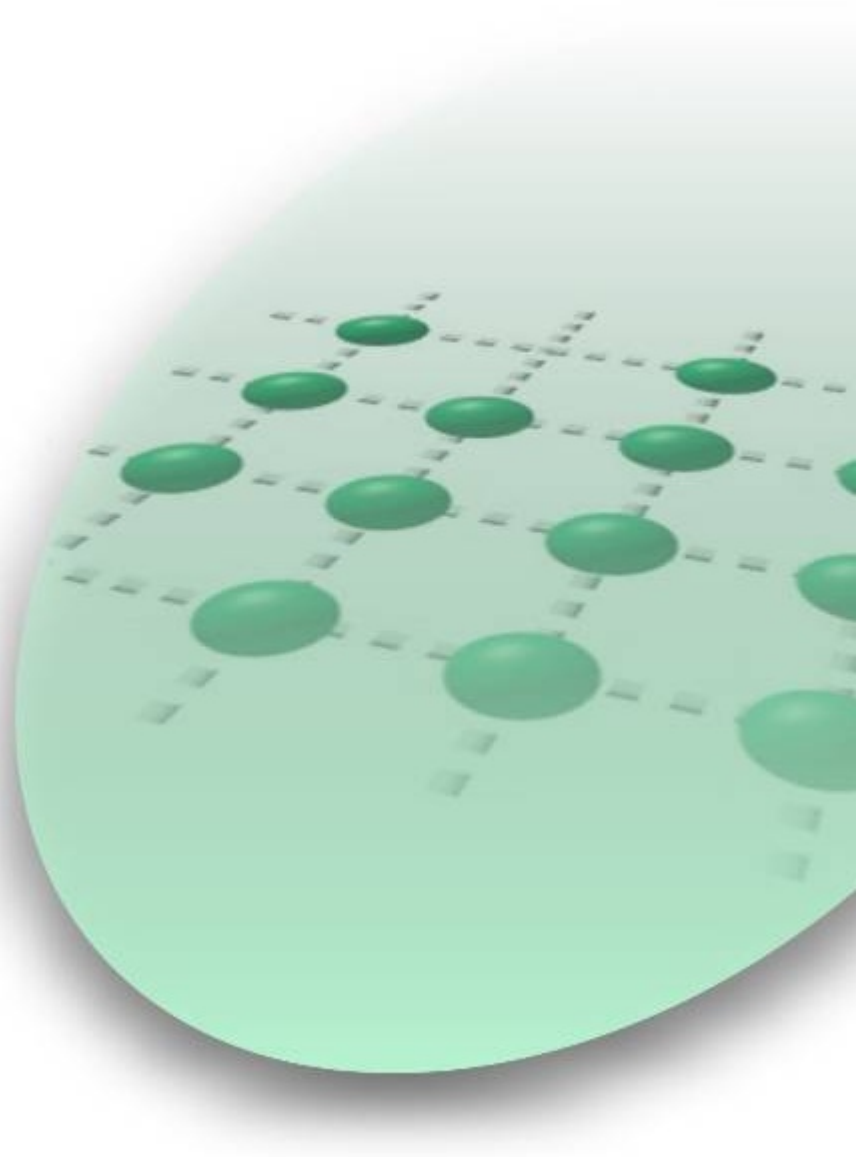
UDP and TCP Communication for Quantum and Premium PLCs running UnityPro Applications

System User Guide

[source code]





33004102.00

Merlin Gerin
Square D
Telemecanique

Schneider Electric
Building a New Electric World

DEC 2006

Contents

Application Source Code 5
Typical Applications 6
System 7
Architecture **7**
PLC and Ethernet Hardware Configuration **9**
Configuration Software **10**
Software Installation **11**
Implementation **14**
Configuring the IPCom services **14**
Installing Function Blocks in UnityPro **18**
Sample Applications **19**
Operate Sample Application on the PLC **21**
Understanding the Sample Application **23**
Appendix 28
Detailed Component List **28**
Concept Version of IPCom **29**
Contact 30

Introduction

This document is intended to provide a quick introduction to the described System.

It is not intended to replace any specific product documentation. On the contrary, it offers additional information to the product documentation, for installing, configuring and starting up the system.

A detailed functional description or the specification for a specific user application is not part of this document. Nevertheless, the document outlines some typical applications where the system might be implemented.

Abbreviations

Word / Expression	Signification
AC	Alternating Current
Advantys	SE product name for a family of I/O modules
Altivar (ATV)	SE product name for a family of VSDs
CANopen	Name for a communications machine bus system
CB	Circuit Breaker
CoDeSys	Hardware-independent IEC 61131-3 programming software
ConneXium	SE product name for a Family of Transparent Factory devices
DC	Direct Current
EDS	Electronic Data Sheet
E-OFF, E-STOP	Emergency Off switch
Harmony	SE product name for a family of switches and indicators
HMI	Human Machine Interface
I/O	Input/Output
IclA (ICLA)	SE product name for a compact drive
Lexium/Lexium05/LXM	SE product name for a family of servo-drives
Magelis	SE product name for a family of HMI-Devices
MB - SL	SE name for a serial Modbus communications protocol
Micro	SE product name for a middle range family of PLCs
NIM	SE product name for a Network Interface Module
Osiswitch	SE product name for a family of position switches
PC	Personal Computer
PDO	Process Data Object (CANopen)
Phaseo	SE product name for a family of power supplies
PLC	Programmable Logic Computer
Powersuite	An SE software product for configuring drives
Premium	SE product name for a middle range family of PLCs
Preventa	SE product name for a family of safety devices
PS1131 (CoDeSys)	SE Product name for PLC programming software with CoDeSys
PS	Power Supply
RPDO	Receive Process Data Object (CANopen)
SE	Schneider Electric
SDO	Service Data Object
Sycon	SE product name of a Field bus programming software

Word / Expression	Signification
Telefast	SE product name for a series of distributed I/O devices
TesysU	SE product name for a decentralised I/O System
TPDO	Transmit Process Data Object (CANopen)
Twido	SE product name of a middle range family of PLCs
TwidoSoft	SE product name for a PLC programming software
TwidoSuite	SE product name for a PLC programming software
Unity (Pro)	SE product name for a PLC programming software
Vijeo Designer	An SE software product for programming Magelis HMI devices
VSD	Variable Speed Drive
WxHxD	Dimensions : Width, Height and Depth
XBT-L1000	An SE software product for programming Magelis HMI devices
Zelio	SE product name for a low range PLC family
ZelioSoft	SE product name for a PLC programming software

Application Source Code

Introduction

Examples of the source code used to attain the system function as described in this document can be downloaded from our „Village“ website under **this** link.

The example source code is in the form of configuration, application and import files. Use the appropriate software tool to either open or import the files.

Extension	File Type	Software Tool Required
AIW	Configuration File	Advantys
CNF	**C**onfiguratio**n** **F**ile	Sycon
CO	**C**AN**o**pen definitions file	Sycon
CSV	**C**omma **S**eparated **V**alues, Spreadsheet	Twidosoft
CTX		Unity
DCF	**D**evice **C**onfiguration **F**ile	Advantys
DIB	**D**evice **I**ndependent **B**itmap	Sycon
DOC	**Doc**ument file	Microsoft Word
DOP	Project File	Magelis XBTL
EDS	**E**lectronic **D**ata **S**heet – Device Definition	Industrial standard
FEF	Export file	PL7
GSD	EDS file (**G**eraete **S**tamm **D**atei)	Profibus
ISL	**Isl**and file, project file	Advantys
PB	**P**rofi**b**us definitions file	Sycon
PDF	**P**ortable **D**ocument **F**ormat - document	Adobe Acrobat
PRO	Project File	PS1131 - CoDeSys
PS2	Export file	Powersuite export file
RTF	**R**ich **T**ext **F**ile - document	Microsoft Word
SPA	Schneider Product Archive	TwidoSuite
STU	Project file	Unity studio
STX	Project file	PL7
TLX	Project file	Twinline control tool
TWD	Project file	TwidoSoft
VDZ	Project file	Vijeo Designer
XEF	Export file	Unity Pro
XPR	Project file	TwidoSuite
ZM2	Project File	Zeliosoft

Typical Applications

Introduction

Here you will find a list of the typical applications, and their market segments, where this system or subsystem can be applied:

- The IPCom communication library is intended for TCP or UDP communication between Schneider Electric Quantum or Premium PLCs and non-Schneider Electric devices and controllers.
- The communication library is available on Quantum and Premium controllers running Unity and also on Quantum controllers running Concept.
- The present document only describes the Unity integration. For detailed documentation of the FactoryCast HMI configuration extension and a detailed communication function block description please consult the documentation delivered with the IPCom package.

Application	Description	Image
Steelworks	IPCom library used for communication to: - B&R HMI systems - Siemens PLCs	
Labeling Machine	IpCom library used for communication to: - Siemens Scada system - Siemens PLCs	

System

Introduction

The system chapter describes the architecture, the dimensions, the quantities and different types of components used within this system.

Architecture

General

The IPCom communication capabilities are achieved using a collaboration of specific communication modules with specific firmware extensions and specific function block libraries for use in the application programs running on the controllers.

The maximum communication capacity for one communication module is:

- 16 TCP connections, bi-directional send and receive, maximum 1460 bytes per telegram, maximum 8000 Bytes altogether
- 16 UDP connections, unidirectional send or receive, maximum 2040 bytes per telegram, maximum 8000 Bytes altogether

Components

Hardware:

In addition to the standard Unity PLC hardware, a FactoryCast HMI communication module is needed:

- for Quantum PLCs: 140 NWM 100 00
- for Premium PLCs: TSX WMY 100

Software:

In addition to the standard UnityPro installation, the following software and libraries are needed:

- FactoryCast HMI Configuration Software
- IPCom function block libraries for IP and TCP communication

IMPORTANT

FactoryCast HMI communication modules used for IPCom Ethernet communication cannot be used for the standard FactoryCast HMI services. A module performing one or more standard FactoryCast HMI services cannot perform IPCom communication.

Quantities of Components

For a complete and detailed list of components, the quantities required and the order numbers, please refer to the components list at the rear of this document.

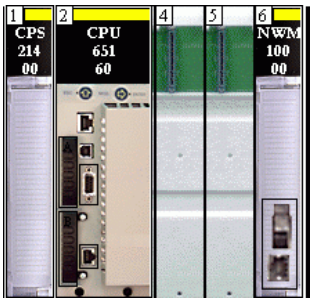
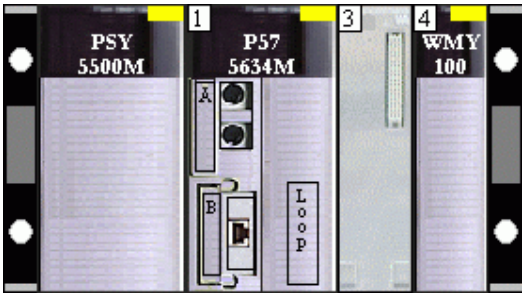
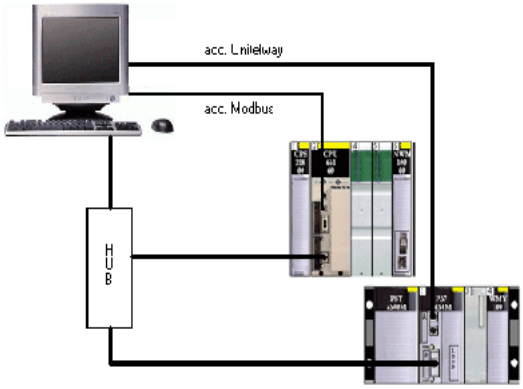
Degree of Protection

Not all the components in this configuration are designed to withstand the same environmental conditions. Some components may need additional protection, in the form of housings, depending on the environment in which you intend to use them.

For environmental details of the individual components please refer to the list in the appendix of this document and the appropriate user manual.

PLC and Ethernet Hardware Configuration

General

Quantum	In addition to standard Quantum PLC hardware a dedicated FactoryCast HMI communication module **140 NWM 100 00** is required.	
Premium	In addition to standard Quantum PLC hardware a dedicated FactoryCast HMI communication module **TSX WMY 100** is required.	
Ethernet Network	In addition to the Ethernet connection a dedicated **Modbus** or, respectively, **Unitelway** programming connection might be necessary.	

Configuration Software

General

The configuration and programming of a PLC with IPCom communication is done with UnityPro and FactoryCast HMI Configuration Software. The IPCom communication package does not need any additional Software tools except that delivered with the IPCom package.

To use the software packages, your PC must have the appropriate Microsoft Windows operating system installed:

- Windows 2000 *or*
- Windows XP.

The software tools are installed to the following default paths:

- UnityPro
  C:\Program Files\Schneider Electric\Unity Pro
- FactoryCast HMI
  C:\Program Files\Schneider Electric\Factory Cast HMI
- IPCom Configuration Software
  Manual installation – described below
- IPCom runtime software
  Manual installation – described below
- Communication Function Blocks
  Manual installation – described below

IPCom Documentation

To date, the documentation for IPCom is only available in German.

- Copy the contents of the IPCom package to a temporary directory, e.g. "**C:\Temp**\"
- The IPCom documentation can be found at

  ..\Ipcom-vx.y\doc

- The doc-subdirectories contain the documentation files:

  konfig
  tcpcom
  udpcom

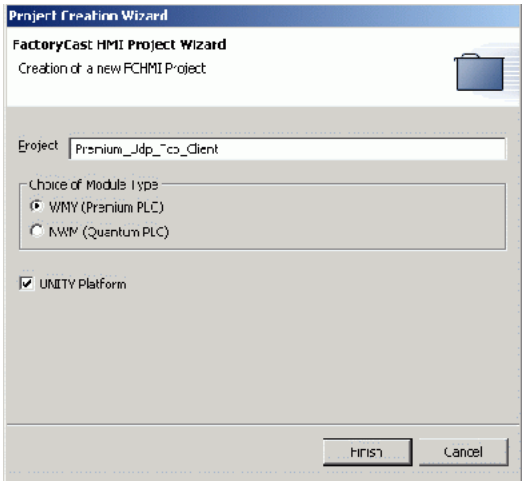
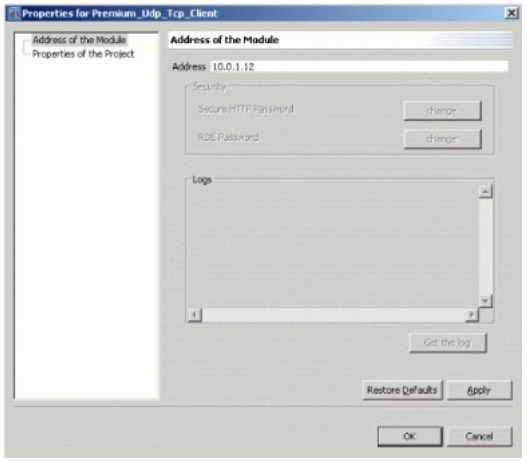
Software Installation

Preconditions

Before continuing with the set up please check that:

- UnityPro is properly installed.FactoryCast HMI configuration software is properly installed.
- The communication modules are equipped with the appropriate firmwareThe PLC configuration, including the communications module IP configuration, is done.
- The communication module is setup with a clean FactoryCast HMI configuration **(Module/Format the Module)**.
- The module is connected to the Ethernet network and running.

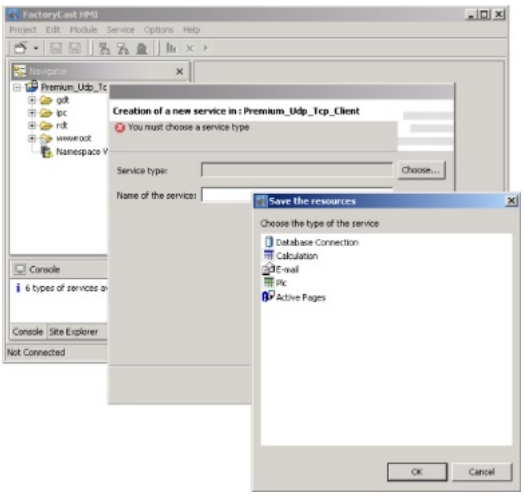
Check Available Services

1. Start the FactoryCast HMI configuration software. 2. Create a new configuration: **Project/New/Project** 3. Set **Project** Name, **Module Type** and **Platform** 4. Commit the project creation via **Finish**. The new project will be created, initialized and shown in the **Navigator** pane.	
5. In the **Navigator** pane, click on the new project and select **Properties** via the context menu. 6. Set the **IP-Address** of the module inside the dialog. 7. Commit with **OK**.	

8. Verify the available services via **Project/New/Service**. click on **Choose**.

9. There will be **no** IPCom service type available.

10. Exit the dialogs with **Cancel**



Install IPCom configuration plugin into FactoryCast HMI Configuration software

1. Close the FactoryCast HMI configuration software

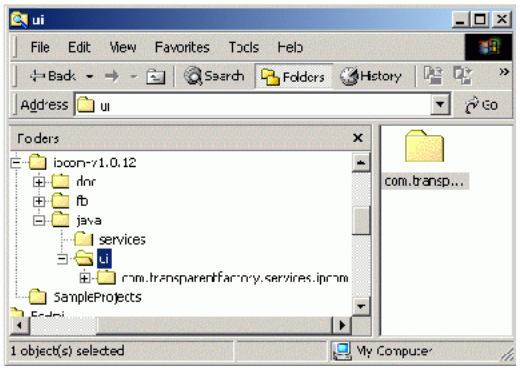
2. and go to the Microsoft explorer.

3. Copy the contents of the IPCom package to a temporary directory, e.g.

   C:\Temp

4. Find the IPCom configuration plug-in at

   ..\Ipcom-vx.y\java\ui



5. Find the plug-in installation path of FactoryCast HMI at

   C:\Program Files\Schneider Electric\Factory Cast HMI\plugins

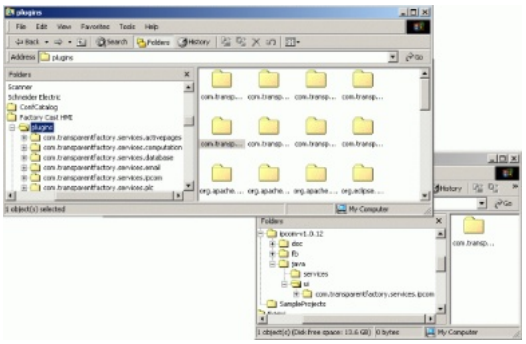
6. Drag the contents of the

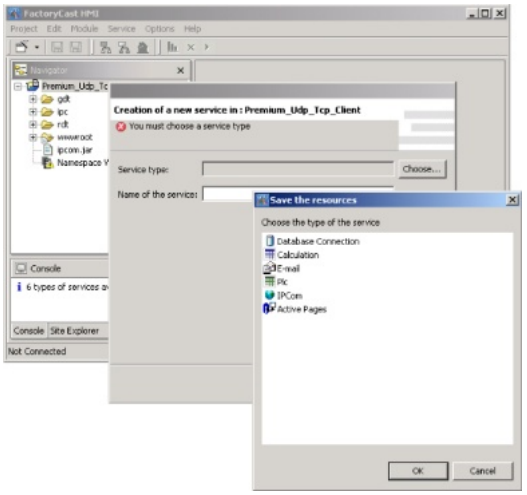
   ..\ui-directory

   to the

   ..\plugins-directory.

7. Start FactoryCast Configuration software.



8. Verify the available services via **Project/New/Service**. Click on **Choose**. 9. **IPCom** service type is now available. 10. Exit the dialogs with **Cancel**.	

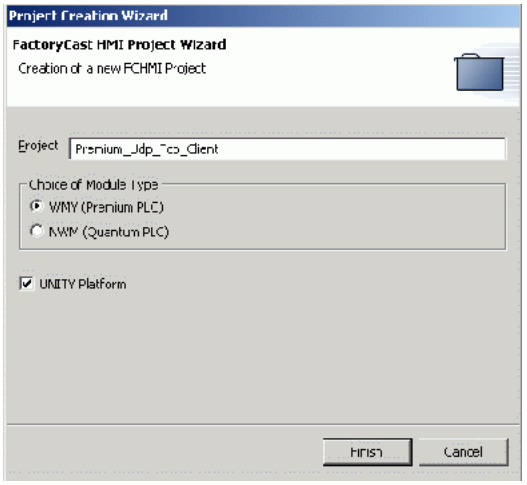
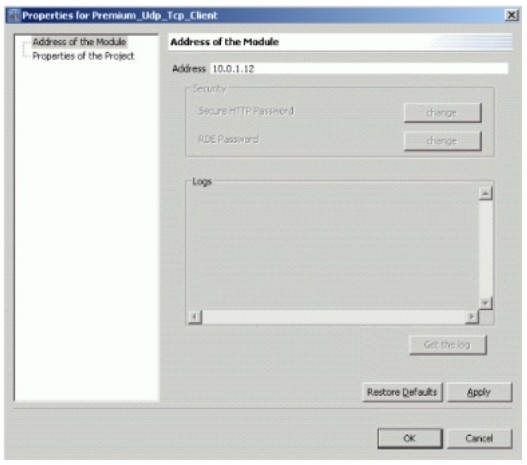
Implementation

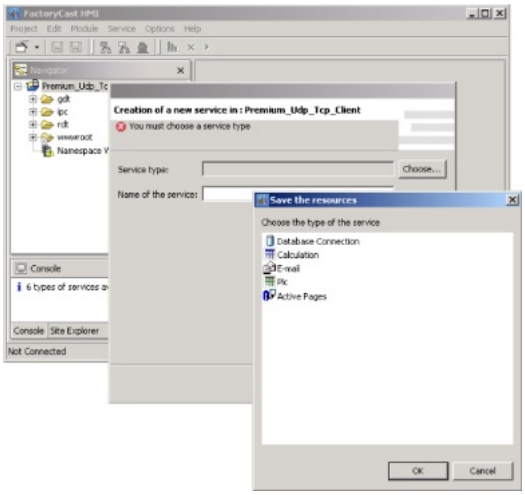
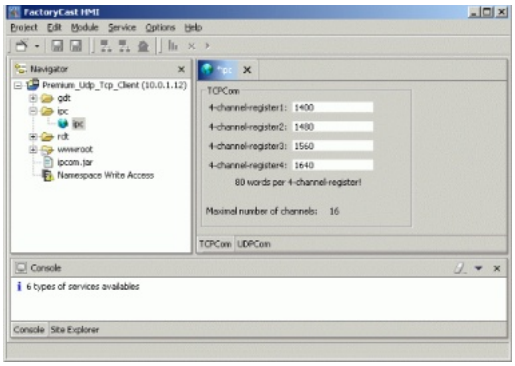
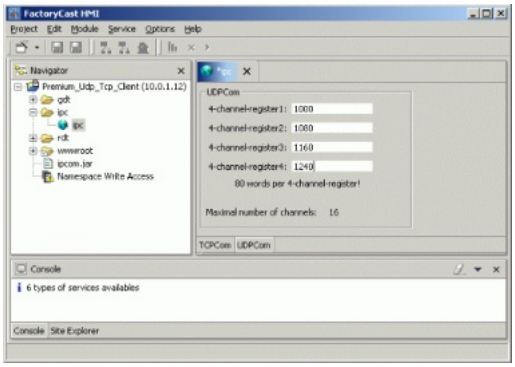
Introduction

The implementation chapter describes all the steps necessary to initialise, to configure, to program and start-up the system to achieve the application functions as listed below.

Configuring the IPCom services

IPCom Configuration in Factory Cast

1. Start FactoryCast HMI configuration software. 2. Create a new configuration: **Project/New/Project** 3. Set **Project** Name, **Module Type** and **Platform** 4. Commit the project creation with **Finish**. The new project will be created, initialized and shown in the **Navigator** pane.	
5. In the **Navigator** pane click on the new project and select Properties via the context menu. 6. Set the **IP-Address** of the module inside the dialog. 7. Commit it with **OK**.	

8. Create the IPCom services with **Project/New/Service**. Click on **Choose** to select the service type. 9. Select the **IPCom** service type. 10. Commit it with **OK.** 11. Choose a Name for the service. 12. Exit with **Finish**.	
13. The new service appears as an element of **IPC** service type. Open the service configuration dialog with a double click. 14. Configure the data exchange areas in the PLC's located memory area for TCP communication capabilities. Set the start address(es) of the 80 Word data area(s). A gradual reservation for 4, 8, 12 or 16 communication channels is supported.	
15. Similarly for UDP, configure the data exchange areas in the PLC's located memory area for the UDP communication. Set the start address(es) of the 80 Word data area(s). A gradual reservation for 4, 8, 12 or 16 communication channels is supported. 16. Close the **IPC** configuration dialog by clicking on the tab's close button. 17. Commit and save with **Yes**.	

Install IPCom service on the FactoryCast HMI module

1. Copy the contents of the IPCom package to a temporary directory, e.g.

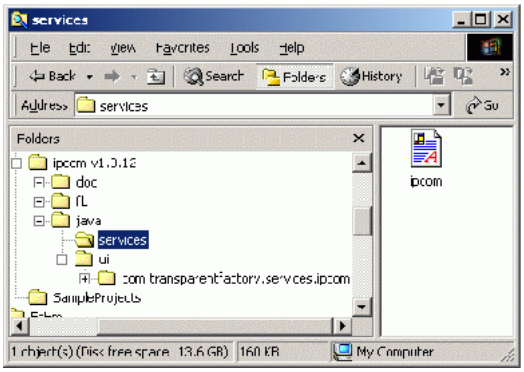
   C:\Temp

2. Find the IPCom runtime software at

   ..\Ipcom-vx.y\java\services

3. The runtime software is in the file

   ipcom.jar



4. Find the service installation path on the FactoryCast HMI communication module in the FactoryCast HMI configuration tool on the **Navigator** pane.

5. The ipcom-service has to be located directly in the **root path of the project**, parallel to the **Namespace Write Access** file.

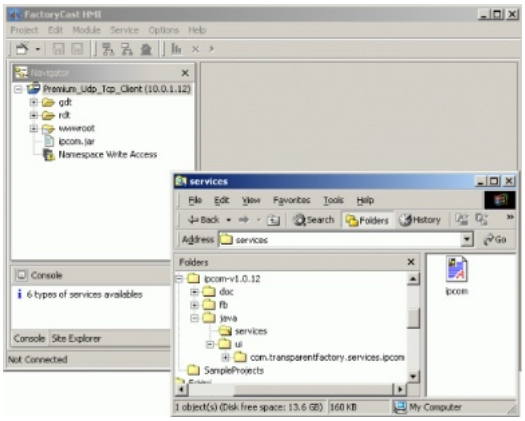
6. Drag the **ipcom.jar** file from the **..\services**-directory on the PC hard disk to the **project-path** inside the **Navigator**.

7. **Download** the service onto the FactoryCast HMI **module**. This has to be done **explicitly**. For this, select the **ipcom.jar** file inside the **Navigator** and do:

   Partial Transfer / PC -> Module

   from the context menu.

8. You will be asked to reboot the module after download. Confirm with **YES.**

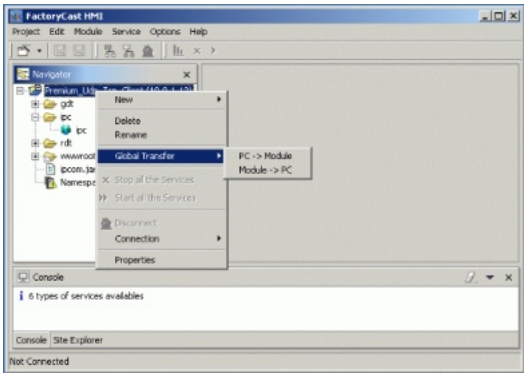


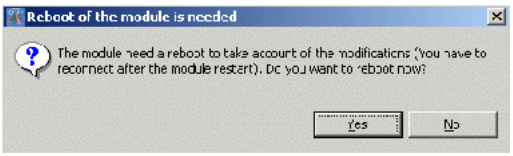
Transfer the configuration to the communication module

1. Transfer the complete configuration to the communication module.

   On the context menu of your FactoryCast HMI project select:

   Global Transfer->PC -> Module



2. You will be reminded again that a module reboot is needed. Confirm the reboot with **YES**. 3. After the reboot the module is ready to operate the communication services requested from the application Program. You can now close or minimize the FactoryCast HMI.	

Installing Function Blocks in UnityPro

Installing or updating IPCom function blocks in the UnityPro application

The IPCom communication function blocks have to be installed in an already created UnityPro project, see the description below. (Alternatively a Custom Library can be created and added to UnityPro with the Types Library Update tool.)

1. Copy the contents of the IPCom package to a temporary directory, e.g.

   C:\Temp

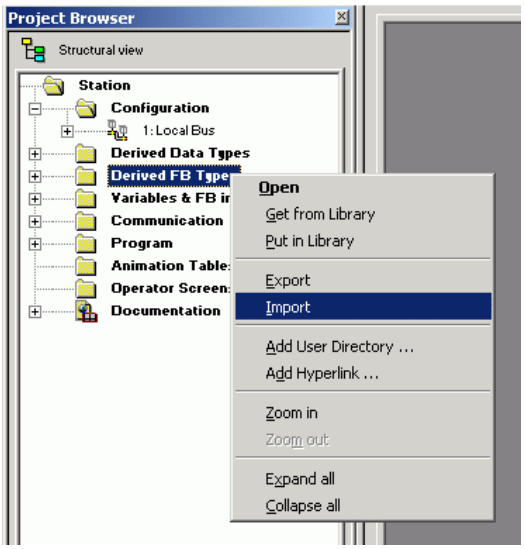
2. The IPCom function blocks library can be found in:

   ..\Ipcom-vx.y\fb\unity\tcpcom
   and
   ..\Ipcom-vx.y\fb\unity\udpcom.\

3. Install the IPCom function block library into your Unity project.

4. Do this for each of the four IPCom function blocks.

5. If you load the complete example project, no extra function block import is necessary as the IPCom function block library is already included in this project.

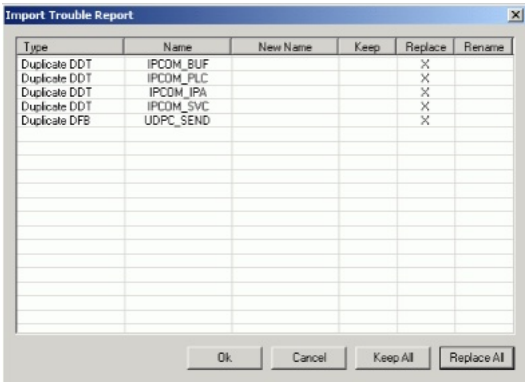


6. If an older version of the IPCom function block library is already installed you can update it to the newer version.

7. In the **Import Trouble Report** dialog click on **Replace All**. All checks appear in the **Replace** column of the dialog.

8. Select **OK** to import the function block.

9. Do this for each of the four IPCom function blocks.

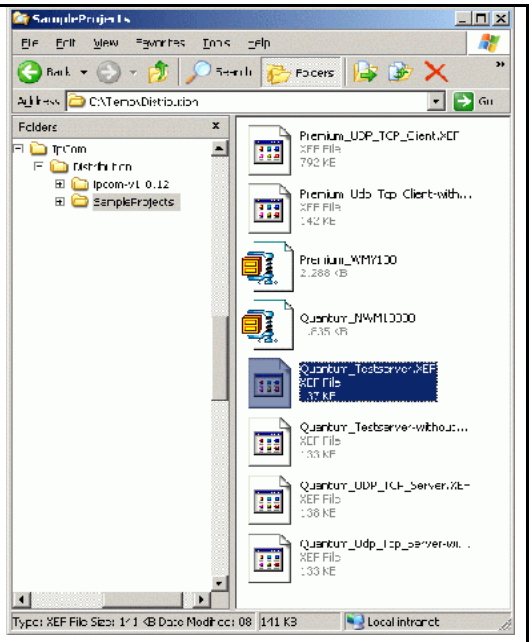


Sample Applications

Introduction

The **SampleProjects** subdirectory of the IPCom distribution contains 8 files:

1. Unity export file **Premium_UDP_TCP_Client.XEF**.
2. The same as 1 but without a PLC rack configuration for easy adaptation to any PLC.
3. A FactoryCast HMI module configuration export file **Premium_WMY100.ZIP.**
4. A FactoryCast HMI module configuration export file **Quantum_NWM10000.ZIP.**
5. A Unity export file **Quantum_Testserver.XEF**.
6. The same as 5 but without the PLC rack configuration for easy adaptation to any PLC.
7. A Unity export file **Quantum_UDP_TCP_Server.XEF.**
8. The same as 7 but without a PLC rack configuration for easy adaptation to any PLC.



Unity Server project file

The example project implements:

1. A TCP server on port 2002. The server immediately returns the data extracted from the received request to the responding client. This can be tested with the PC application **tcpclient.exe**.
2. A UDP server on port 2001. The server immediately returns the data extracted from received request to the responding client. This can be tested with the PC application **udpclient.exe**.

It is this application example which is described below.

Quantum_Testserver.XEF
together with
PC application **tcpclient.exe**
and
PC application **udpclient.exe**

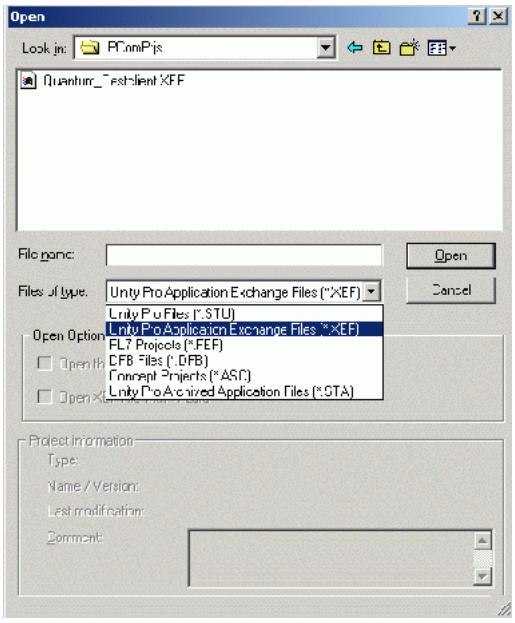
Unity Peer-to-Peer communication project files	The example projects implement: 1. A Quantum with a TCP server on port 2002 and a UDP server on port 2001 with similar functionality as described above. 2. A Premium with a TCP client and a UDP client communicating with the Quantum server PLC. An additional Operator Screen allows convenient control and monitoring of the communication clients. **No further description for these examples is included here.**	**Premium_UDP_TCP_Client.XEF** together with **Quantum_UDP_TCP_Server.XEF**
FactoryCast HMI module configuration export files	The communication module export files contain the configuration used for all the examples mentioned above. These are not needed if the module configuration is done manually.	**Premium_WMY100.ZIP** and **Quantum_NWM10000.ZIP**

Operate Sample Application on the PLC

Operate sample application

1. Launch **UnityPro** on your PC.
2. Open the sample application Quantum_Testserver.XEF.

   Before locating the project file in your file system, select the appropriate file type **.XEF**.
3. Click on **Open** once the file is selected.

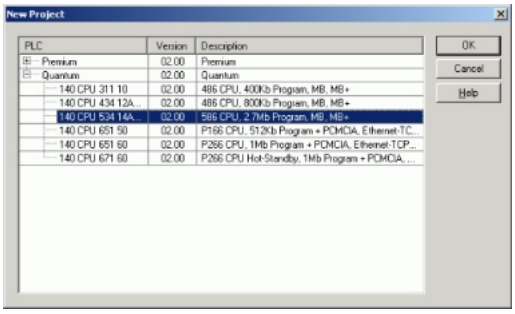


4. If your PLC configuration is not similar to the PLC configuration in the sample project, it might be better to configure part of the application manually to fit your PLC.

   To do this open the sample application:

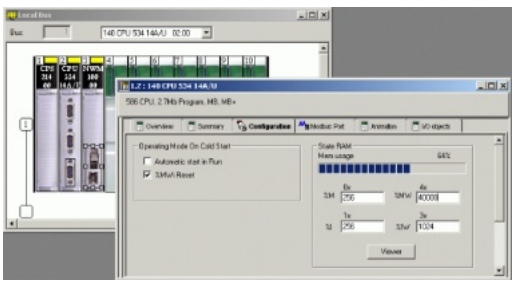
   Quantum_Testserver-withoutConf.XEF.

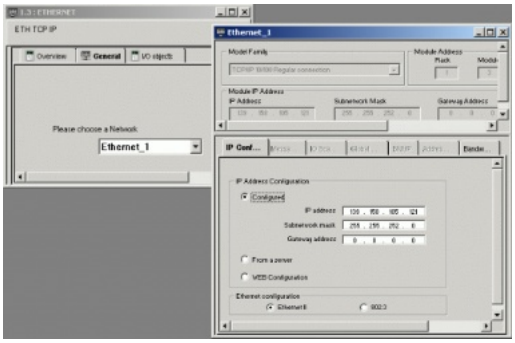
   You are asked to select a PLC. **T**he application is a Quantum application so you have to select an appropriate **Quantum PLC**.



5. Complete the rack configuration. Adjust the **backplane** size and add at least a **power supply** and the communication module

   140 NMW 100 00.
6. Adjust the **State Ram** memory usage to 40000 words.

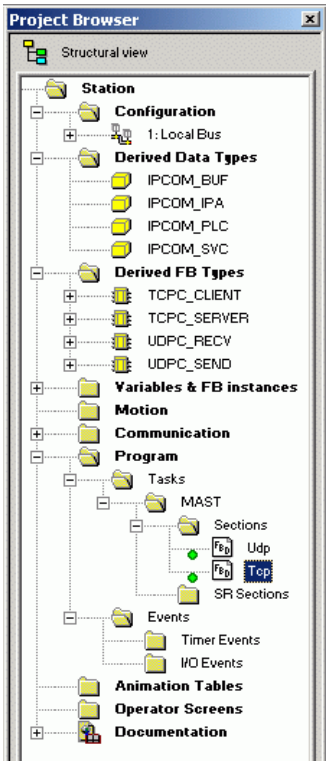


7. **Link** the communication module to the network. 8. Adjust the **IP address** as required.	
9. **Build** the complete project. 10. **Connect** to your PLC. 11. **Transfer** the project to your PLC. 12. Start the PLC with **Run**. The status bar should indicate **EQUAL**, **RUN**, **UPLOADINFO OK** and **BUILD**.	

Application program main parts

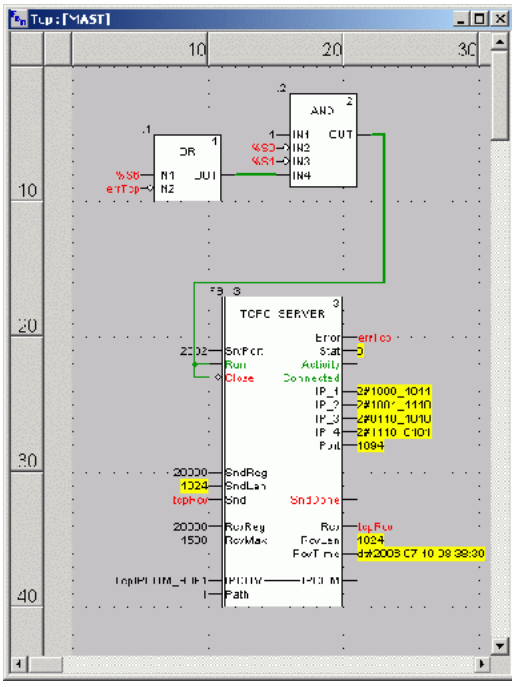
Opening the sample project creates all relevant objects for the application. The UnityPro **Project Browser** displays these objects:

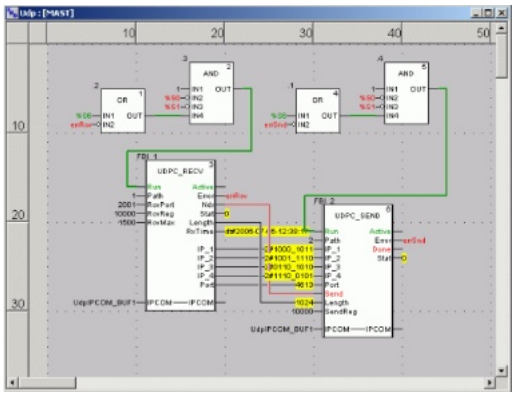
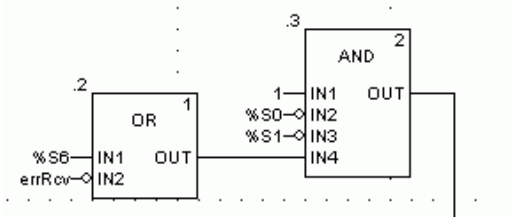
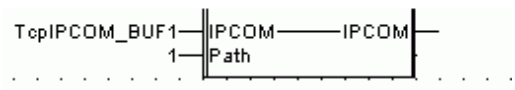
- The entry **Derived Data Types** contains the data type definitions used for variables intended to accomplish the access to the communication channels.
- The entry **Derived FB Types** contains the definitions for the 4 function blocks managing the communication data exchange. These DFBs use the data types defined in the node above.
- The entry **Program** contains the two FBD sections of the application program. The section **Tcp** implements the TCP server function, the section **Udp** implements the UDP server function.



Section Tcp

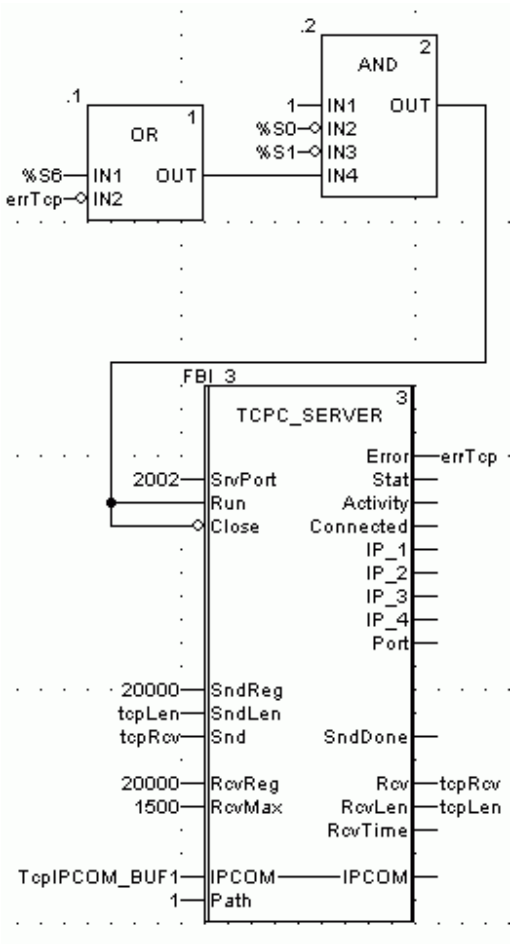
The TCP server is implemented in the function block TCPC_SERVER. The logic blocks connected are concerned with the startup and error recovery behaviour of the function block.



Section Udp	The UDP server is implemented in the two function blocks UDPC_RECV and UDPC_SEND. The logic blocks connected to theses are concerned with the startup and error recovery behaviour of the function blocks.	
Common parts: **Startup and error recovery**	• The communication FBs are disabled during the cold start (**%S0**) and warm start (**%S1**) scans. Enabling after these scans restarts the FBs. • The communication FBs are disabled when detecting errors (**Error** output). As long as the error exists the FBs are disabled but enabled periodically every second (**%S6**) until they can re-start regular operation.	
Common parts: **Fault indication**	• **Error** and **Stat** indicate errors detected by the FB. This could be a configuration error (wrong configuration of IPCom on FactoryCast HMI, wrong parameterization of FB, communication problems during operation). Identified errors are specified with an error code.	
Common parts: **Collaboration with FactoryCast HMI module**	As seen during the FactoryCast HMI configuration before, each module is capable of executing 16 TCP and 16 UDP services. The services are organized into maximum channels with 4 services on each channel. One channel is capable of executing one type of service, mixing services is not supported on channels. • The **IPCOM** In-Out-parameter must point to a variable of data type IPCOM_BUF. This is the management data structure for one channel. • The **Path** parameter has to address one of the 4 communication services. This is the path inside the channel.	

TCP Server FB parameters

- SvrPort is set to 2002.
- Run and Close are connected to the startup and error recovery logic.
- SndReg and RcvReg are set to the same register address. This ensures that the received data is immediately available for the response.
- SndLen is set according to the length of the received data indicated on RcvLen.
- Snd is triggered from Rcv when a request is received.
- RcvMax determines the maximum size of the receive buffer in memory in bytes.
- Activity indicates a running transaction (request, response).
- Connected indicates that a client has connected to the server. The IP-address (IP_1 .. IP_4) and send port (Port) of the client are indicated.
- SndDone indicates that the response is finished.
- RcvTime indicates the time at which the request was received.



Test of section Tcp

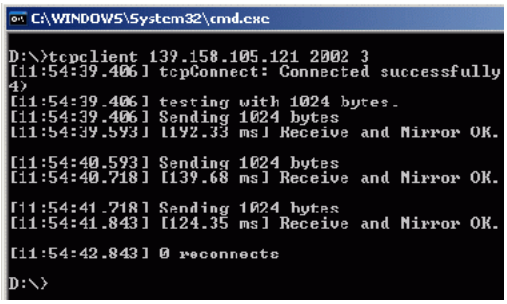
To test of the TCP server function, the command line tool **tcpclient.exe** is used.

1. Copy the contents of the IPCom package to a temporary directory, e.g.
   C:\Temp
2. Find the TCP test client at
   ..\Ipcom-vx.y\tools

The test client requires at least three parameters:

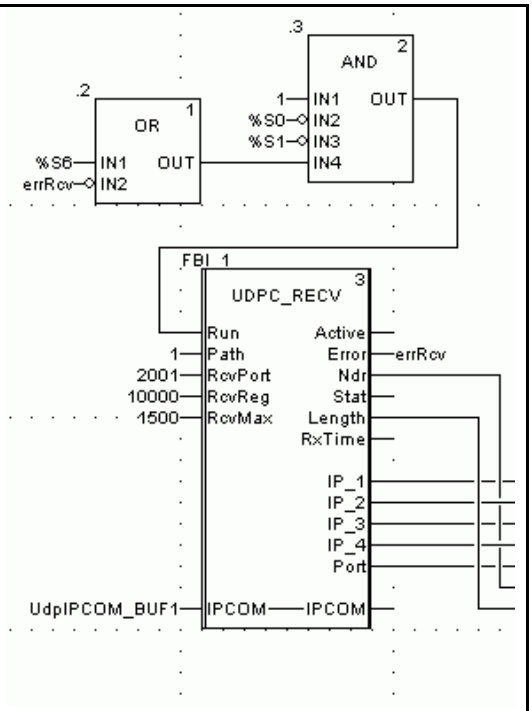
- The IP-address of the TCP-server on the PLC/communication module.
- The port on which the server listens.
- The repetition rate.

Invocation of the tool without parameters produces a short help information.


C:\WINDOWS\System32\cmd.exe
D:\>tcpclient 139.158.105.121 2002 3
[11:54:39.406] tcpConnect: Connected successfully
4>
[11:54:39.406] testing with 1024 bytes.
[11:54:39.406] Sending 1024 bytes
[11:54:39.593] [192.33 ms] Receive and Mirror OK.
[11:54:40.593] Sending 1024 bytes
[11:54:40.718] [139.68 ms] Receive and Mirror OK.
[11:54:41.718] Sending 1024 bytes
[11:54:41.843] [124.35 ms] Receive and Mirror OK.
[11:54:42.843] 0 reconnects
D:\>

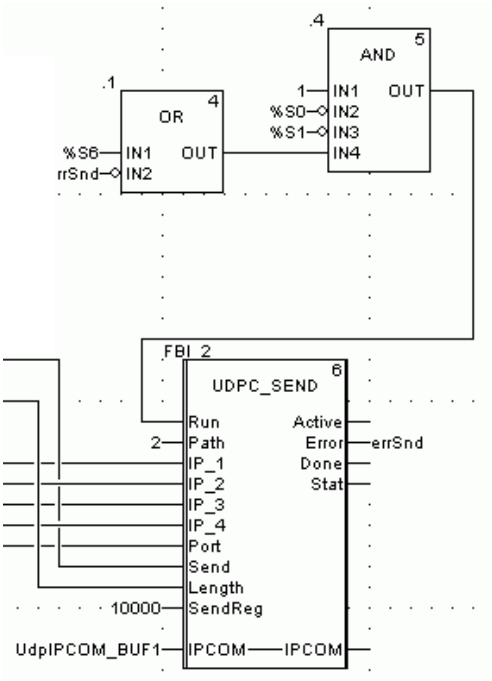
UDP Receive FB parameters

- Run is connected to the startup and error recovery logic.
- RcvPort is set to 2001.
- RcvReg points to the start of the receive buffer in memory.
- RcvMax determines the maximum size of the receive buffer in memory in bytes.
- Active indicates a running receive operation.
- Ndr indicates that valid data has been received.
- Length indicates the length of the received data in bytes.
- RxTime indicates the time at which the data was received.
- IP_1 .. IP_4 and Port indicate the IP-address and send port of the sender.



UDP Send FB parameters

- Run is connected to the startup and error recovery logic.
- IP_1 .. IP_4 and Port are set to the IP-address and port to which the send FB is ordered to transfer data. In this example the IP-address and send port from sender are used. It is expected that a listener on the sender's IP-address is listening on a receive port with the port number of the send port of the preceding send operation.
- Send triggers the send operation. It is activated by the receive FB's Ndr. This means that the data is immediately sent back after received.
- Length sets the data length to be transmitted. It is set by the receive FB's Length output. This means that all the data received is to be sent back.
- SndReg points to the start of the send buffer in memory. It is set to the same values as the receive FB's RcvReg and points to the same receive buffer. This means that the received data is sent back.
- Active indicates a running send operation.
- Done indicates that the send operation is finished.



Test of section Udp

To test the UDP server function the command line tool **udpclient.exe** is used.

1. Copy the contents of the IPCom package to a temporary directory, e.g.

   C:\Temp

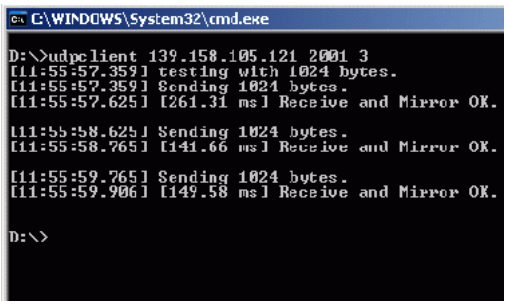
2. Find the UDP test client at

   ..\Ipcom-vx.y\tools

The test client requires at least three parameters:

- The IP-address of the UDP-server on the PLC/communication module.
- The port on which the server listens.
- A repetition rate.

Invocation of the tool without parameters gives a short help information.


C:\WINDOWS\System32\cmd.exe
D:\>udpclient 139.158.105.121 2001 3
[11:55:57.359] testing with 1024 bytes.
[11:55:57.359] Sending 1024 bytes.
[11:55:57.625] [261.31 ms] Receive and Mirror OK.
[11:55:58.625] Sending 1024 bytes.
[11:55:58.765] [141.66 ms] Receive and Mirror OK.
[11:55:59.765] Sending 1024 bytes.
[11:55:59.906] [149.58 ms] Receive and Mirror OK.
D:\>

Appendix

Detailed Component List

Additional Hardware-Components				
Pos.	**Amt.**	**Description**	**Part Number**	**Rev./ Vers.**
1		Ethernet communication module for Quantum PLCs 140 NWM 100 00	140 NWM 100 00	
or				
2		Ethernet communication module for Premium PLCs TSX WMY 100	TSX WMY 100	

Additional Software-Components				
Pos.	**Amt.**	**Description**	**Part Number**	**Rev./ Vers.**
1		FactoryCast HMI Configuration Software	TLX CD FCHMI V1M	V1.2
2		IPComm function block libraries for UDP and TCP communication		

Concept Version of IPCom

Concept

For Concept systems with Quantum PLCs the IPCom TCP and UDP communication is also available.

The required Ethernet communication module 140 NWM 100 00 is applied as described above.

For the Concept application program a set of similar Concept function blocks is available.

Contact

Author	Telephone	E-Mail
Schneider Electric GmbH CMachines and Process Architectures	+49 6182 81 2555	cm.systems@de.schneider-electric.com

Schneider Electric GmbH
Steinheimer Strasse 117
D - 63500 Seligenstadt
Germany

As standards, specifications and designs change from time to time, please ask for confirmation of the information given in this publication